FAKULTA ELEKTROTECHNIKY
A KOMUNIKAČNÍCH TECHNOLOGIÍ
ústav telekomunikací

Multimediální systémy
Cvičení č. 4

Garant kurzu: **doc. Ing. Petr Číka PhD.**

Cvičící: **Ing. David Kohout**
Ing. Milan Bubniak

Akademický rok: 2022/2023

Osnova

- Spring MVC
- **REST** (Representational State Transfer)
- Úvod do šablonování webových stránek ve Springu
- Engine **Thymeleaf**

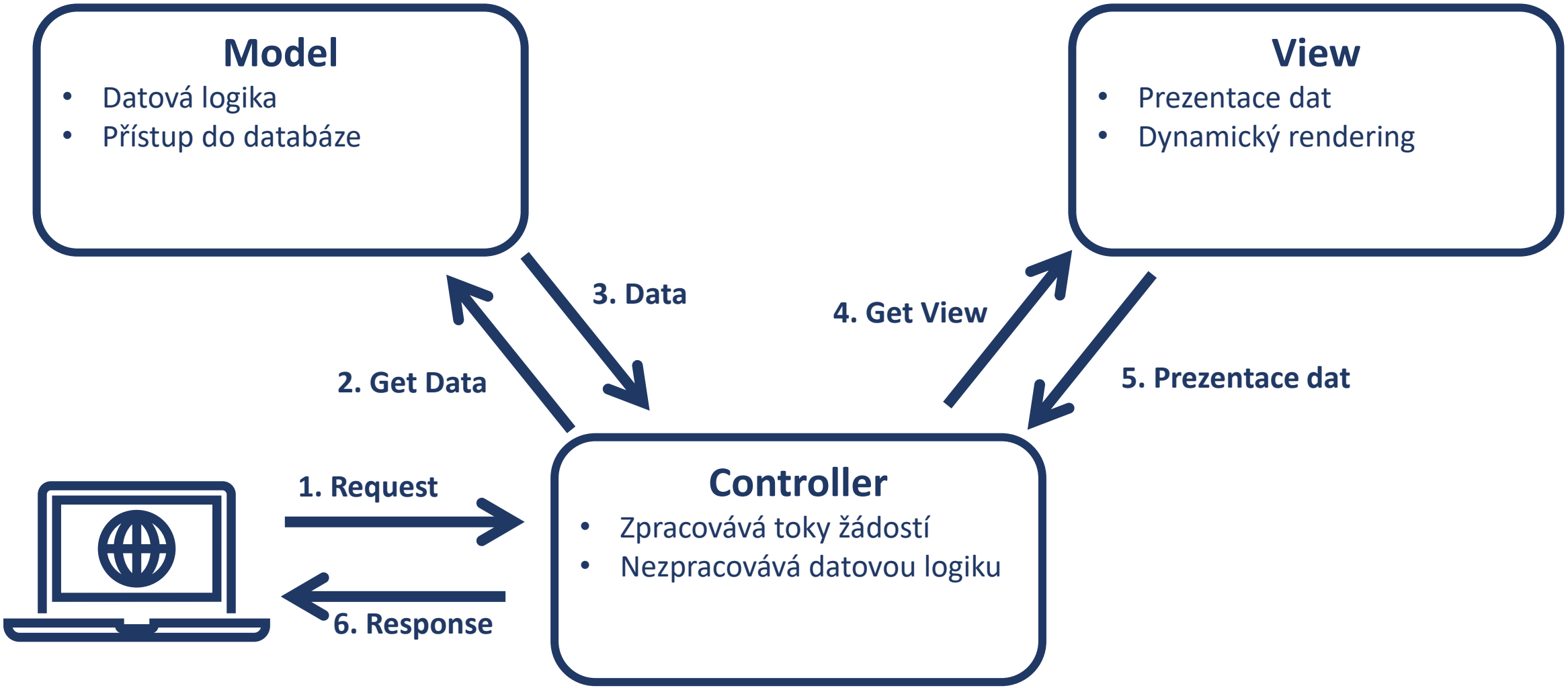
Spring MVC

- MVC = Model-View-Controller
- Jeden z přístupů pro tvorbu webových aplikací, rozdělení do vrstev/částí
- Každá část se stará jen o to co má. To usnadňuje vývoj a údržbu

- Model – stará se o data a jejich získávání
- View – zajišťuje reprezentaci dat (výsledek vidí uživatel)
- Controller – řídí vše, předává žádosti

- Model a View přímo nikdy nekomunikují – vše jde přes Controller

MVC


Model
• Datová logika
• Přístup do databáze
View
• Prezentace dat
• Dynamický rendering
3. Data
2. Get Data
4. Get View
5. Prezentace dat
1. Request
6. Response
Controller
• Zpracovává toky žádostí
• Nezpracovává datovou logiku

REST

- **REST** (Representational State Transfer)
- Představen v roce 2000
- Obsahuje čtyři základní metody označené CRUD (**Create** – vytvoř, **Retrieve** – získej, **Update** – změň a **Delete** – vymaž)

- POST (vytvoř), GET (získej), DELETE (vymaž), PUT (změň)

POST

- Neboli **create** – vytvoření dat
- Není známý přesný identifikátor zdroje (ve chvíli volání ještě neexistuje). Využívá domluvený identifikátor: „**endpoint**“
- Metodou POST je předáván parametr s konkrétní hodnotou.

```
curl -u uzivatel:heslo -d „user=bob" https://example.cz/api/user/
```

Tímto příkazem je vytvořen zdroj „**bob**“.

Přepínač –d = data POST request

GET

- Klasický požadavek na webovou stránku

```
GET /rest/user/bob
Host: www.example.cz
```

- Existuje unikátní zdroj identifikovaný URI. Pomocí metody GET získáme data tohoto zdroje.

```
curl https://www.example.cz/api/user/bob
```

DELETE

- Příkaz na mazání zdroje pomocí volání URl zdroje

  ```
  DELETE /rest/user/bob
  Host: www.example.cz
  ```

- V praxi není rozšířeno
- Servery většinou mají omezení pouze na POST a GET
- U serverů nepodporující DELETE je většinou nahrazena metodou POST, kde serveru sdělíme, že chceme provést metodu DELETE

```
curl -u uzivatel:heslo --http-request DELETE https://example.cz/api/user/bob
```

PUT

- Operace změny
- Modifikace existujících záznamů
- Přidání nových záznamů, pokud neexistují

```
curl -X PUT -d "user=bob" -d "user=alice" https://example.cz/api/user/bob
```

Prostor pro dotazy

Úvod do šablonování webových stránek

1. Vytvoření metody se statickým HTML
2. Vytvoření statické šablony HTML
3. Vytvoření šablony HTML s podporou dynamického obsahu

Vytvoření metody se statickým HTML

```
@GetMapping("form")
@ResponseBody
public String helloForm() {
  String html = "<html>" +
        "<body>" +
        "<form method='post' action='hello'/>" +
        "<input type='text' name='name'/>" +
        "<input type='submit' value='Pozdrav!'/>" +
        "</form>" +
        "</body>" +
        "</html>";
  return html;
}
```

Statická šablona HTML

1. V projektu Spring za pomocí engine Thymeleaf
2. Vytvoření HTML souboru v adresáři „**templates**“

   <Kořenová složka projektu>/src/main/resources/templates/
3. V controlleru vytvořit metodu pro předání HTML souboru

Dynamická šablona HTML

1. V projektu Spring za pomocí engine Thymeleaf
2. Vytvoření HTML souboru v adresáři „**templates**“

   <Kořenová složka projektu>/src/main/resources/templates/
3. V controlleru vytvořit metodu pro předání HTML souboru

Oproti statické šabloně lze do dynamické vkládat proměnné, které lze v průběhu měnit.

Engine Thymeleaf

- Moderní engine pracující na straně serveru, který má na starost šablonování
- Vhodný pro HTML5 JVM webový vývoj
- Poslední verze 3.0.11
- Webová stránka projektu: https://www.thymeleaf.org/
- Integrace v IDE: Eclipse a IntelliJ IDEA
- Licence: Open source – Apache License 2.0
- GitHub: https://github.com/thymeleaf

Příklad Thymeleaf

- Šablona pro Thymeleaf musí obsahovat:
  ```
  <html lang="cs" xmlns="http://www.w3.org/1999/xhtml" xmlns:th="http://www.thymeleaf.org">
  ```
- Proměnné z modelu lze vytáhnout pomocí Thymeleaf elementů.
  Tyto elementy začínají většinou jako **th:**
- th:href – pro dynamicky (v šabloně) měněný odkaz
- th:text – nastavení textu v kombinaci s proměnnou modelu
- th:each – for each cyklus v šabloně
- th:field a th:action – pro získávání dat z formulářů
- th:if a th:unless – varianta if-else

Spoločná ukázka

Samostatný úkol

- V třídě WebController vytvořte 3 metody, které budou splňovat následující:
  - Metoda myself bude naslouchat na adrese myself. Bude použita statická šablona, kde bude napsáno vaše jméno a id. Obojí bude v elementu <h1> a jméno bude červeně a ID tučně v kombinací s jinou barvou dle vaší volby.
  - Další metody budou alice a bob, každá bude naslouchat na jiné adrese(alice, bob)
  - Obě metody budou využívat jednu dynamickou šablonu (template.html) a 2 rozdílné css soubory, se jmény odpovídající metodám (tzn. alice.css a bob.css)
  - *Alice* bude napsána jako <h1>, kurzívou a červeně
  - Bob bude taktéž v <h1>, podtrženě a modrou barvou
  - Css lze dynamicky měnit: `th:href="@{'/css/' + ${name} + '.css'}"`